Activity Recognition: Translation across Sensor Modalities Using Deep Learning

Tsuyoshi OKITA
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata,
Kitakyushu-shi
Fukuoka, 804-8550, JAPAN
tsuyoshi.okita@gmail.com

Sozo INOUE
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata,
Kitakyushu-shi
Fukuoka, 804-8550, JAPAN
sozo@mns.kyutech.ac.jp

Abstract

We propose a method to translate between multi-modalities using an RNN encoder-decoder model. Based on such a model allowing to translate between modalities, we built an activity recognition system. The idea of equivalence of modality was investigated by Banos et al. This paper replaces this with deep learning. We compare the performance of translation with/without clustering and sliding window. We show the preliminary performance of activity recognition attained the F1 score of 0.78.

Author Keywords

Activity Recognition, Deep Learning, Multi-modality

ACM Classification Keywords

H.1.2 [User/Machine Systems]

Introduction

The architecture of deep learning may facilitate the handling of multiple modalities for activity recognition in the sense that the architecture to learn representation is the overlapping aim for each modality. One interesting application of such representation is translation between different modalities. Banos et al. [2] proposes a translation between the 3 dimensional signals of Kinect data (3D position) and the 3 dimensional signals of the Inertial Measuring Units (IMU) data (3D acceleration).

UbiComp/ISWC' 18 Adjunct, October 8–12, 2018, Singapore, Singapore.
© Association for Computing Machinery.
ACM 978-1-4503-5966-5/18/10...$15.00.
https://doi.org/10.1145/3267305.3267512

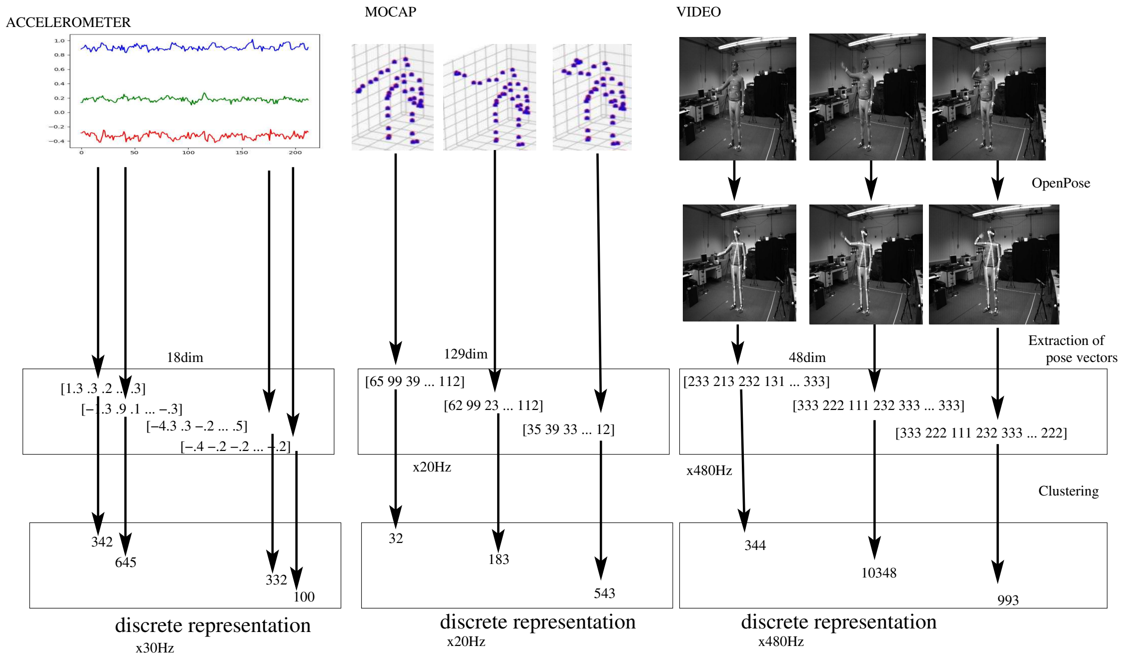

Figure 1: Figure shows a schema of handling spatio-temporal aspect of multi-modal data in MHAD dataset[11]. Signals are converted into real-valued vectors (or word embedding) which is then used as discrete representation (or word surface form) in encoder-decoder architecture. Note that 18, 129, and 48 dimensions are specific in our experiments. Note also that this schema shows the case without sliding window. In the case of sliding window depending on the sliding window size and window gap, some data with overlapped region will be used multiple times.

In deep learning, translation reminds us of the encoder-decoder architecture [4, 17, 1, 7] for Machine Translation between the source and the target representations. Among various extended works an image caption generation system [21] replaces the source side with *object* representation in image. This paper aims to translate from one modality into another modality.

Using such translation systems, it is possible to set up activity recognition system in unsupervised learning. Luo et al. [6] proposes a convolutional LSTM-based encoder-decoder only with two hidden states thus simply used as prediction, with down-/up-sampling networks.

Our contribution is twofolds: (1) new translation system from one modality to another and (2) new activity recognition system in unsupervised learning.

pref para	freq
None	587
-100	9239
-1000	8216
-5000	7133
-1x10^4	6684
-5x10^4	5768
-1x10^5	5111
-1x10^6	1612
-1x10^7	780
-5x10^7	153
-5x10^8	3122
-1x10^9	9982

Figure 2: Table shows the size of cluster by affinity propagation [5] with different preference parameter. Dataset is the motion capture side in MHAD dataset.

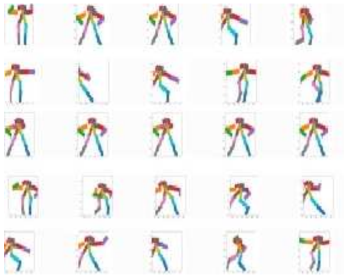

Figure 3: Figure shows an example of pose representation extracted by OpenPose library.

Preprocessing: Representation of Signals

We first explain how the raw format of spatio-temporal data is converted into the input representation (Refer Figure 1): notably (1) how the spatio-temporal signal is converted into discrete representation, i.e. word representation in encoder-decoder architecture (where temporal aspect is handled by LSTMs by encoder-decoder), and (2) how these discrete representation is further processed as window-based representation (where some temporal aspect is encoded in the embedding). Without loss of generality, we describe our examples of multi-modal signal using three kinds: accelerometer, mocap, and camera.

The first step is to transform signals into discrete representation and do a normalization. Note that some data are already in the form of discrete representation. For example, accelerometer signal takes the form of 3 dimensional spatio-temporal volume: spatial representation which has 18 dimension consisting of 3-axis accelerometers which are attached to 6 locations of body, and temporal representation which is 30Hz. 18 dimensional real-valued vector is considered as the representation of accelerometer. In the case of video signals, the video is transformed to the 48 dimensional pose vectors by OpenPose library [3] which we describe details in the subsection.

The second step is to obtain the input representation for the encoder-decoder architecture by clustering. First, we deploy clustering algorithm to obtain clusters. The input for the conventional encoder-decoder architecture of MT is a pair of word sequences. Using a pair of word sequences using 1-of-K coding scheme, word embedding is obtained as optimisation progresses. In our case, we may use the real-valued vector (in the middle raw in Figure 1) as representation for this purpose. However, due to the smallness of the dataset with extremely high-dimensional space, there are virtually no overlapping among elements whose real-valued vectors are the same. In this reason, we do clustering in order to obtain the elements whose properties are similar. We use affinity propagaption [5] for this purpose. Most clustering algorithm needs to supply the number of cluster beforehand, this affinity propagation clustering solves the appropriate cluster number as a result. The number of cluster is related to the preference parameter while when this parameter is not given the algorithm seeks the optimised number of clusters (Ref Table 2). Second, based on the results of clustering, we allocate the word identity number for the real-valued vectors. We treat such word identity number as the word surface form. We expected that in the case when the parallel corpus is sufficiently big, such clustering of embedding would not be necessary. It turned out that under small data (MHAD data) training did not converge without clustering when the target side had large complexity. Third, we use the real-valued vector as the pretrained embedding to encoder-decoder architecture avoiding calculation based on the word surface form.

The third step is optional. When the sliding window approach is not taken, we use the results of the step two. Otherwise, we append the end of real-valued vector the corresponding real-valued vectors within the sliding window. Then, we apply the clustering for this appended data.

Video Preprocessing

We preproces video input to extract its *pose* representation using OpenPose [3]. While many methods are related to background separation/segmentation before extracting the representation, an OpenPose-based extraction can capture the human pose avoiding background separation. Cons is that if more than two persons appeared in the video, we need to identify correctly which person is our target in the latter method. In this sense, we have several alternatives to extract *human* representation from video inputs but we did not try them.

Our Method

Figure 4 shows our overall architecture. We use the input representation described in the previous section as the input z.[1] The mapping from z (one modality) to y (another modality) can be considered as the translation systems, while the mapping from z (input signals) to u (activity label) can be considered as the activity recognition systems. For the former, this is a combination of the encoder-decoder architecture and the multi-scale feature pyramid. For the latter, based on the output signals of encoder-decoder architecture, the multi-instance learner classifies this into one of the activity labels.

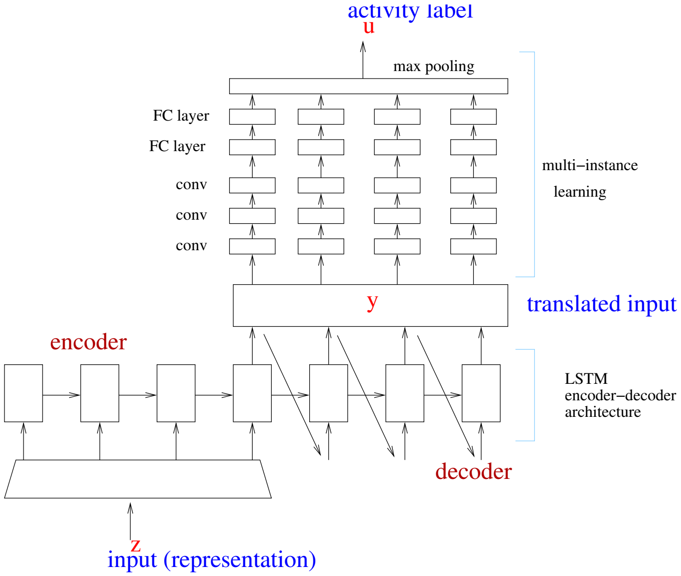

Figure 4: Figure shows the overall architecture of our method.

Encoder-Decoder Architecture

The RNN encoder-decoder architecture consists of (a) a RNN encoder which computes a representation c for each source sentence and (b) a RNN decoder which generates one target word at a time and hence decomposes the conditional probability. More formally, let $x_i (= \{x_1, \ldots, x_n\})$ denote a source sentence[2] and $y_i (= \{y_1, \ldots, y_m)$ denote a target sentence corresponding to x_i $(0 \leq i \leq N)$. Then a encoder-decoder architecture calculates the conditional probability $p(y|x)$ of translating a source sentence to a target sentence, as in (1):

$$\log p(y|x) \quad = \quad \sum_{j=1}^{N} \log(y_j | y_{<j}, c) \tag{1}$$

The RNN encoder computes the current hidden state h_j given the previous hidden state h_{j-1} and the current input x_j, as in (2):

$$h_{enc;j} \quad = \quad \text{RNN}_{\text{enc}}(x_j, h_{enc;j-1}), \tag{2}$$

where RNN$_{\text{enc}}$ denote a RNN unit, generating a representation c for each source sentence $\{x_0, \ldots, x_t\}$. The RNN decoder computes the current hidden state h_j given the previous hidden state with a representation c.

$$\left\{ \begin{array}{lcl} h_{dec;j} & = & \text{RNN}_{\text{dec}}(h_{dec;j-1}, c), \\ p(y_j | y_{<j}, c) & = & \text{softmax}(g(h_{dec;j})) \end{array} \right. \tag{3}$$

where RNN$_{\text{dec}}$ denotes a conditional RNN unit. The cost function is $\frac{1}{M} \sum_{\ell=1}^{M} (\log p_\theta(y_n | x_n))$ where M is the size of parallel corpus.

[1] Although we handle multi-modal signals, they are allocated the word ID by cluster number. We can handle these as if signals are words (Ref Figure 1).

[2] In Figure 4, x is the output of multi-scale feature pyramid, which is then an input to the encoder-decoder architecture.

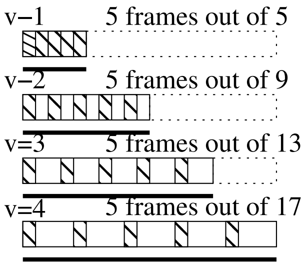

Figure 5: Figure explains the multi-scale feature pyramid when v=1, 2, 3, and 4.

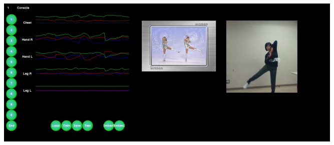

Figure 6: Figure shows the acquisition mode of data logger used for collecting the UFO data.

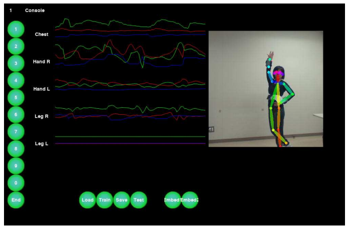

Figure 7: Figure shows the play mode of data logger used for collecting the UFO data.

Global Attention
The global attention model [7] considers the history of alignment mapping in a global manner, considering the covering of the alignment between the source and the target words. The alignment vector a_y is a variable length whose size equals the number of time steps on the source side.

$$\begin{aligned} a_y(x) &= \text{align}(h_{dec}, \overline{h}_{enc}) && (4) \\ &= \frac{\exp \text{score}(h_{dec}, \overline{h}_{enc})}{\sum_{x'} \exp(\text{score}(h_{dec}, \overline{h}_{enc'}))} && (5) \end{aligned}$$

The score is a content-based function as follows.

$$\text{score}(h_{dec}, \overline{h}_{enc}) = \begin{cases} h_{dec}^T \overline{h}_{enc} \\ h_{dec}^T W_a \overline{h}_{enc} \\ v_a^T \tanh(W_a[h_{dec}, \overline{h}_{enc}]) \end{cases} \quad (6)$$

where initial alignment a_t is calculated by softmax$(W_a h_{dec})$.

Length Attention
Since the length of the source side will approximately determine the length of the target side, it is natural to impose the length constraint on the target side. This is since the rate of the sensor signal of one modality and that of another modality is consistent as far as the frequency of both modalities do not change. We implement the length attention mechanism in the following way. The information about the length is provided in the form of an additional input to the RNN_{dec}. Thus, the following embedding in Equation (7) is supplied to the decoder.

$$\ell_i = \begin{cases} 0 & (l_i \leq f_{length}(y_i)) \\ l_i - f_{length}(y_i) & \text{(otherwise)} \end{cases} \quad (7)$$

Multi-Scale Feature Pyramid
We adopted the assumption that the sequence of frames across modalities is synchronized and sampled at a given temporal step v and concatenated to form a spatio-temporal 3-d volume (a dynamic pose; [10]). We move the value v to leverage multiple temporal scales in encoder-decoder architecture, accommodating differences in tempos/styles of articulation of different uses. Regardless of the step v we can use the same number of frames at each scale. This is called a multi-resolution spatial pyramid. In our case, we need to handle the input as well as the output, which is not related to this, although the synchronization of input and output may not be necessary.

Multiple Instance Learning
The multiple instance learning (MIL) framework [20, 19] is applied greedily after the output of decoder. For given bags $\{Y_i\}$ which contains a number of instances $\{y_{ij}\}$ and labels $\{U_i\}$ corresponds to the bag $\{Y_i\}$, a multi-class MIL is to seek the classification rule: $Y \longrightarrow U$.

Let $\{y_{i1}, \ldots, y_{iN}\}$ be the i-th sequence of output by encoder-decoder (corresponding to the i-th input sequence). Now, we do not see this as a sequence but as a bag of instances $B_i = \{u_{ij}\}$. Then we assign all the labels of the frame to the bag so that each bag has the label $U_i = \{u_{in}\}$ where $u_{in} = 1$ indicates the presence of activity class i.

We take an approach of convolutional neural network towards MIL[20, 19]. We use three convolutional layers followed by two fully-connected layers. A batch normalization is added after each convolutional layer and the ReLU activation function is used for all layers. We view the output as independent posterior probability estimates for each class. In order to assign a label of the maximum scoring instance

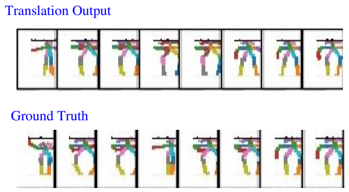

Figure 8: Example of translated results and ground truth (acc to camera direction).

window size	#seg
s01-a01-r01 . . . s01-a11-r01	55
s02-a01-r01 . . . s02-a11-r01	55
s12-a01-r01 . . . s12-a11-r01	55
total	660

Figure 9: Figure shows the details of segments of MHAD dataset. Missing data of mocap are s04-a08-r05 and s10-a05-r05, those of accelerometer are s02-a07-r05 and s04-a08-r05, and those of cam are c01-s02-a03-r01, c01-s09-a05-r05, c02-s04-a08-r05, c02-s06-a06-r05, c02-s06-a07-r01, c02-s06-a08-r01, and c02-s06-a08-r02.

to the bag, we use the max pooling layer as in (8):

$$\hat{U}_i \quad = \quad \{\hat{u}_{in}\} = \{\max_j f_n(y_{ij})\} \qquad (8)$$

where $f_n(y_{ij})$ is the predicted probability of class n on instance y_{ij}. For the multi-class MIL, we use the cross entropy loss summed over all the classes, as in (9).

$$J_i \quad = \quad -\sum_n (u_{in} \log \hat{u}_{in} + (1 - u_{in}) \log(1 - \hat{u}_{in})) \; (9)$$

Experimental Results

Experimental Settings

We used two datasets. The first one is our in-house dataset named UFO data set while the second one is called the MHAD dataset from Berkeley[11][3]. The UFO data have two modalities: video and accelerometer. This data contains the recordings of dance of UFO and daily activities such as walking and sitting down. The length of this data is around 48 hours. This data do not have labels, consisting of activities of three persons. Due to this reason, this dataset is only used as looking at the perplexity. The camera position for video was fixed in front of the targeted person. Three-axis accelerometer were attached in five locations in the body (both hands and legs and chest).

For the MHAD datset, we used video, accelerometer, and mocap modalities among others. We used video with Cluster-01/Cam01-02 subsets, and the whole mocap (optical) and accelerometer data with 12 persons/5 trials (Ref Figure 9). Video input was preprocessed by OpenPose library as in the same way as the UFO dataset. Optical mocap inputs have the position of the keypoints whose dimension was 129. Accelerometer inputs were placed in 6 places in the body whose dimension was 18.

	freq	corpus size	acc2moc	moc2acc
V	30Hz	0.6k/30/27	60023.32	200.12
VC	30Hz	0.6k/30/27	19.74	18.04
SW_5	6Hz	30k/150/150	10.73	10.72
SW_{10}	3Hz	15k/100/82	11.28	11.01
SW_{20}	1.5Hz	7.4k/100/96	13.67	13.73
SW_{30}	1Hz	4.8k/100/100	15.03	21.35
SW_{60}	0.5Hz	2.5k/50/50	35.98	29.03
			acc2cam	cam2acc
V	30Hz	1.24k/30/30	14700.22	1.20
VC	30Hz	1.24k/30/30	2.60	4.40
SW_5	6Hz	59k/500/500	4.39	4.48
SW_{10}	3Hz	29k/500/500	3.21	4.48
SW_{20}	1.5Hz	14k/500/500	3.40	3.14
SW_{30}	1Hz	9k/500/500	3.33	4.49
SW_{60}	0.5Hz	4.5k/250/250	3.94	4.49

Table 1: Figure shows the perplexity on test set for translation from acc → moc, moc → acc, acc → cam, and cam → acc. V, VC, and SW denote the variable length approach, the variable length with clustering approach, and the sliding window with clustering approach. SW_{10} means the window size of 10.

We used the parameters in the RNN encoder-decoder with word embedding size 500, rnn modules 500, dropout 3, maximum sentence length 400, and batch size 100. We used $v = 2, 3, 4$ for multi-resolution spatial pyramid. We use Titan Xp.

Results

For the UFO dataset, the training finished with perplexity 65.02 (cam to acc) and 760.33 (acc to cam). The perfor-

[3] http://tele-immersion.citris-uc.org/berkeley_mhad.

	freq	corpus size	cam2moc	moc2cam
V	30Hz	1.24k/30/30	47000.40	9600.50
VC	30Hz	1.24k/30/30	3.43	12.29
SW_5	6Hz	29k/250/210	10.96	4.01
SW_{10}	3Hz	14k/400/400	7.94	3.45
SW_{20}	1.5Hz	7k/200/197	9.80	3.40
SW_{30}	1Hz	4.5k/200/197	10.4	3.55
SW_{60}	0.5Hz	2.9k/90/80	25.10	4.09

Table 2: Figure shows the perplexity on test set for translation from cam → moc and moc → cam. V, VC, and SW denote the variable length approach, the variable length with clustering approach, and the sliding window with clustering approach. SW_{10} means the window size of 10.

mance measured by RMSE [4] was 0.21 (cam to acc) and 0.36 (acc to cam). Figure 8 shows an example of translation outputs while Figures 10 show the representation drawn by t-SNE[9].

For MHAD dataset, experimental results are shown in Tables 1, 2 and 3. For translation we did an experiment in three manners: the variable sequence length approach V, the variable sequence with clustering approach VC, and the sliding window with clustering approach SW. V is an approach taken in traditional Machine Translation while SW is an approach which is often taken by activity recognition. VC can be viewed as Machine Translation with pretrained embedding. For unsupervised activity recognition we did an experiment from camera to accelerometer.

First, we observed a relatively large vocabulary size. Embedding size is equivalent with the vocabulary size. In this case, the embedding size of accelerometer is 152,820, mocap is 293,258, and cam is 148,874. As a reference in MT, the vocabulary of German uses around 80,000 or 50,000. These figures are far bigger than that. The performance on V for acc2moc and acc2cam whose perplexities were more than 100,00, which may be due to this.

Second, we observed that the length of sequence in MHAD dataset is often long. V has often a long sequence which is more than 500 words. The long sequence translation is reported to have one weakness in NMT which is more than 50 words [1]. SW limits itself with the sliding window size, which is fixed as the sliding window size. If this sliding window size is within 50 words, it is expected that NMT performs better. The performance of SW_{20} (moc2cam, cam2acc) and SW_{10} (cam2moc, acc2cam) were better than other window size, which confirmed this.

Third, SW has obvious weakness in the point that it splits the sentence ignoring the dependencies among different windows. V and VC can capture dependencies among word sequence. Although the complexities are problems in acc2moc and acc2cam, when it works, cam2moc performed better than SW approaches.

Fourth, when the target side was accelerometer in V, the training finished fairly well with low perplexity. When the target side was camera or mocap, the perplexity did not go down. One possible reason for the latter is the small training set of 1.2k or 430 parallel sentences.

For unsupervised activity recognition from camera to accelerometer, F1 was 0.78 in average and accuracy was 0.8 for the camera to accelerometer direction (Refer Table 3).

Background

We target several long-term goals of multi-modal translation/unsupervised activity recognition developed in this paper. First, we anticipate the necessity of incremental

[4] This is since this data does not have labels.

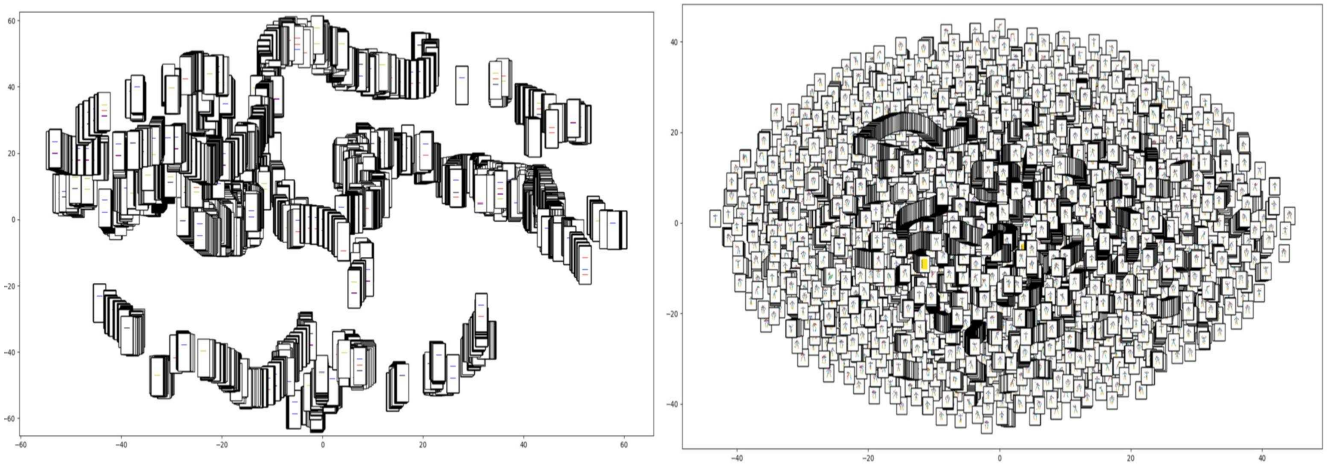

Figure 10: The figure in the upperside shows representation in the camera side drawn by t-SNE while the figure in the lowerside shows representation in the accelerometer side drawn by t-SNE.

	F1	acc
1	1.0	1.0
2	1.0	1.0
3	1.0	1.0
4	1.0	1.0
5	1.0	1.0
6	1.0	1.0
7	0.8	0.6
8	0.5	0.3
9	1.0	1.0
10	0.8	0.6
11	1.0	1.0
ave	0.78	0.80

Table 3: Figure shows the F1 score and accuracy for cam-acc pair (in this direction).

learning of new activities which are not defined before. For example, a chatbot is a conversational agent which exchanges words with human being. We develop a chatbot for telemedicine which identifies the possible diseases for patients[8]. Combined with IoT technology, we want a chatbot to assess the condition of the patient or to make a physical test of the patient at runtime. In order to realize this robot-like chatbot, it needs to acquire new activities as a result of conversation. In this case such new activity may not even have a (reference) name. The characteristics in this domain are that the targeted activities are not predefined, dynamically changing, and relatively large in number.

Second, more importantly, what we seek by translation of one modality to another can be regarded as the high-level contextual abstraction of the low-level sensor configurations. The higher-level contextual abstractions can be semantically replaced or combined according as the need of upper layers. Opportunistic human activity recognition [15] advocates this strategy to build recognition methods for given available sensor data which themselves adapt themselves dynamically. When the sensor data are abundant and controlled rather than scarse, we can choose which sensors to use or which sensors to replace with other sensors. In fact, it is common for human being to use this strategy in our daily life. Human do not deploy all the senses but only deploy a few sense when we are not attentive. But depending on the situation we can deploy more than one sense. For example, when you ride a bike and approach to the corner whose view is bad, adding to the visual perception you will deploy the auditory perception carefully.

Conclusion

We proposed a method to translate between multi-modalities using an RNN encoder-decoder model. Based on such a model allowing to translate between modalities, we built an activity recognition systems. Preliminary experimental results show that for sufficiently large data, this method did work. However, as the complexity of the decoder side increases, this method faced with difficulties for small data.

There are several avenues for further work. First, as was seen from the results, we need to know the cause of difficulties. Since most of the difficulties came from the target side, i.e. camera or motion capture whose complexity was large, a smallness of the dataset is one definite problem. However, this might be solved by the better cleaning strategy, better selection strategy of the data features or data augmentation. We need to investigate how to overcome difficulties in MHAD data. Second, Shi et al. [16] introduced the convolutional LSTM modules in order to reduce redundancy for spatial data. It is interesting to replace our LSTM in encoder-decoder architecture with the convolutional LSTM, together with the comparison with Luo et al. [6]. Third, translation among modalities have two characteristics: (1) the signal order would not basically change, and (2) delay in one side of modality is common. Although the latter is difficult for a traditional sequence classifier but would require NMT, the former suggests that the complexity of standard NMT may not be required but simpler one such as monotonic NMT[14] can be sufficient. It is interesting to apply the monotonic NMT in this situation. Fourth, we mimicked the situation where we always attain the 100% coverage since we train/test in offline manner. We tried to supply all the inputs/outputs when clustering. It is interesting if we can remove this constraint.

Acknowledgements

The authors would like to acknowledge insightful comments and encourangement by anonymous reviewers.

REFERENCES

1. Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, arXiv, 2014.
2. Banos, Oresti and Calatroni, Alberto and Damas, Miguel and Pomares, Hector and Rojas, Ignacio and Sagha, Hesam and del R. Millán, Jose and Troster, Gerhard and Chavarriaga, Ricardo and Roggen, Daniel, “Kinect=IMU? Learning MIMO Signal Mappings to Automatically Translate Activity Recognition Systems Across Sensor Modalities”, ISWC12, pp. 92–99, 2012.
3. Zhe Cao and Tomas Simon and Shih-En Wei and Yaser Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”, CVPR, 2017.
4. Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, Learning Phrase Representations using RNN

Encoder-Decoder for Statistical Machine Translation, EMNLP, 2014.

5. Brendan J. Frey, Delbert Dueck, Clustering by Passing Messages Between Data Points, Science, 315(5814), pp. 972-976, 2007.

6. Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, Li Fei-Fei, “Unsupervised Learning of Long-Term Motion Dynamics for Videos”, CVPR, 2017

7. Minh-Thang Luong, Hieu Pham, Christopher D. Manning, Effective Approaches to Attention-based Neural Machine Translation, EMNLP, 2015.

8. Tittaya Mairittha, Tsuyoshi Okita and Sozo Inoue, Pre-Consulting Dialogue Systems for Telemedicine: Yes/No Intent Classification, WellComp workshop (collocated at UbiComp18), 2018.

9. Laurens van der Maaten and Geoffrey E. Hinton. Visualizing Non-Metric Similarities in Multiple Maps. Machine Learning 87(1):33-55, 2012.

10. Natalia Neverova, Christian Wolf, Graham W.Taylor, and Florian Nebout, “Multi-scale deep learning for gesture detection and localization”, Workshop on Looking at People (ECCV), 2014.

11. Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, Rene Vidal and Ruzena Bajcsy. Berkeley MHAD: A Comprehensive Multimodal Human Action Database. In Proceedings of the IEEE Workshop on Applications on Computer Vision (WACV), 2013.

12. Tsuyoshi Okita, Sozo Inoue, “Recognition of Multiple Overlapping Activities Using Compositional CNN-LSTM Model”, Ubicomp Adjunct, Sep, 2017.

13. Francisco Ordonez, Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. Sensors 16:115, 2016.

14. Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, Douglas Eck, Online and Linear-Time Attention by Enforcing Monotonic Alignments, arXiv, 2017.

15. Daniel Roggen, Gerhard Troster, Paul Lukowicz, Alois Ferscha, Jose del R. Millan, Ricardo Chavarriaga, Opportunistic human activity and context recognition, Computer, 46(2), pp. 36-45, 2013.

16. Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, Wang-chun Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”, arXiv, 2015.

17. Ilya Sutskever, Oriol Vinyals, and Quoc Le, Sequence to Sequence Learning with Neural Networks, NIPS, 2014.

18. Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Yaser Sheikh, “Convolutional pose machines, CVPR, 2016.

19. Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, Wenyu Liu, Revisiting Multiple Instance Neural Networks, arXiv, 2016.

20. Jiajun Wu, Yinan Yu, Chang Huang, Kai Yu, Deep Multiple Instance Learning for Image Classification and Auto-Annotation, arXiv, 2015.

21. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention” arXiv, 2015.

22. Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G.Derpanis, and Kostas Daniilidis, “MonoCap: Monocular Human Motion Capture using a CNN Coupled with a Geometric Prior”, PAMI, 2018.